Connect Four

Connect Four is a 2-player board game, in which the players choose a color and then take turns dropping colored tokens into a seven-column, six-row vertically suspended grid.

The objective of this lab is to create a program that will allow you to play Connect Four against another player or against the computer.

Read and understand the code provided

There is a lot of code already provided. Make sure you read the code, comments and docstrings.

The game board will be represented by a list of lines:

```
grid = [
    line1,
    line2,
    line3,
    # etc...
]
```

Each line is itself a list of "cells". Each cell is a string (one character): it can be empty (for example `-`) or it can have a symbol played by a player (for instance, `X` or `O`).

```
line = ["-", "-", "X", "O", "-"]
```

The grid then has the following format:

```
grid = [
    ["-", "-", "-", "X", "-"],
    ["-", "-", "X", "O", "O"],
    ["-", "X", "O", "O", "O"],
    ["X", "X", "X", "O", "X"],
]
```

is the following grid:

```
   X
  XOO
```

```
 XOOO
XXXOX
```

In this case, player 1 (`X`) won with the diagonal.

You can iterate over the grid and access elements in the grid:

```
# All elements one by one
for line in grid:
    for element in line:
        print(element)

# Each line
for line in grid:
    print(line)

# With column indexes
for line in grid:
    for column_number, element in enumerate(line):
        print("Column", column_number, "is", element)

# If you need line and column indexes
for line_number, line in enumerate(grid):
    for column_number, element in enumerate(line):
        print("Element on line", line_number, "and column", column_number, "is",
element)

# Access element on line 3 (= index 2) and column 5 (= index 4)
element = grid[2][4]
```

The following functions are provided:

- `make_diagonal`: returns a diagonal line based on a starting position in a grid.
- `check_connect4_win`: returns the symbol played by the winning player if there is one, `False` otherwise.
- `make_grid`: returns an empty grid of the dimensions provided.

Complete the missing code

`check_connect4_line`

This function takes a list of characters (or a string) as argument = a line of discs. If the line contains 4 identical elements, return this element (= winning disc). Otherwise, return False.

`make_column`

This function takes a list of lists as input (the grid). It returns a list made of the elements located in the column column_number.

`check_column`

This function returns True if the column index provided belongs to the grid.

`add_symbol_to_grid`

Change the grid: play the symbol at the column index provided. Make sure the symbol "falls down" to the bottom of the column / the last empty cell.

`main`

This is the main function. Improve it by printing additional information about the game, the grid, etc. You can create your own functions if you want too!